905900-292

METHOD FOR MOTION SIMULATION OF AN ARTICULATED FIGURE USING ANIMATION INPUT

CROSS-REFERENCE TO RELATED APPLICATION

This application claims priority pursuant to 35 U.S.C. § 119(e) to U.S. Provisional Application Number 60/426,560, filed November 15, 2002, which application is specifically incorporated herein, in its entirety, by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to methods for animating digitally-modeled articulated figures using a computer.

2. Description of Related Art

An aspect of computer-generated animation involves simulating the motion of an articulated figure in response to externally-applied forces. For example, it may be desirable to cause a character in a computer game or motion picture to react to events, such as a collision with an object or another character, in a realistic way. It is further desirable to achieve this realistic motion in an automatic or semi-automatic fashion, without requiring an animator to manually pose the articulated figure as if reacting to the externally-applied forces. Presently, various methods and tools exist to simulate the response of an articulated figure to external forces, using the equations of motion to calculate the motion of each articulated link of the figure. Once physical parameters of an articulated figure have been defined, the response of the articulated figure to a defined impulse or other applied force can be automatically simulated. The simulated response of the articulated figure may then be used as the basis for generating an animated depiction of the response, in a manner well understood in the art.

Articulated figures for which motion is simulated in this manner are sometimes referred to as "rag dolls," and the response of the figure to the externally applied force indeed resembles the passive and involuntary movements of an inanimate doll. Sometimes this is the desired effect, as when the character is to be depicted as killed or knocked unconscious by a bullet or other blow. Many times, however, it is desirable to depict an animated character that continues to struggle or otherwise move as if under intelligent self-control, while also being moved involuntarily by an externally applied force. For example, it may be desirable to depict a character involuntarily recoiling backwards after landing a punch on an opponent, while continuing to fight.

Animated characters, of course, lack the intelligent self-control needed to generate voluntary muscle movement. Therefore, a character's voluntary movement is normally animated using a human animator, who manually poses the character's articulated "skeleton" through a sequence of key frames. In the alternative, or in addition, motion data is captured from an appropriate subject and transformed into a pose sequence for a modeled articulated figure using motion-capture methods. The manually-defined key frame poses of the figure then become the basis for a finished animation of the character, as known in the art. Animators may be assisted in this process by various tools that apply defined constraints and/or rules of behavior for the articulated figure to fill in between the key-frames, to string predefined animation sequences together in ways that simulate voluntary behavior, and/or to ensure realistic posing in the key frames. Such tools, however, do not generally obviate the need for a least a degree of manual input from an animator, or collection of motion-capture data, to portray basic voluntary character movements.

Skilled animators and motion-capture actors are capable of manually defining a figure's voluntary movements to achieve a variety of desired effects in the final animation. Involuntary motion, however, poses a greater problem for the animator and actor. Many animators find it difficult to intuit realistic involuntary movements of an articulated figure, particularly when involuntary motion is combined with voluntary motion. In the case of motion-capture, it may be too difficult, too dangerous, or simply

impossible to replicate forces and reactions that are desired for an animation sequence. Hence, it is often difficult to realistically animate characters in scenes that require both voluntary and involuntary motion of an articulated figure. Furthermore, any given manually-defined animation that includes both voluntary and involuntary movement, no matter how realistic, can only depict a single response. If it is desired to depict a response with a different involuntary motion component, another manual animation process is required, even if the voluntary component of the motion is unchanged.

The pose of an articulated figure is defined by a set of parameters that describe transforms between nodes of a hierarchical data structure, which is sometimes called a directed acyclic graph. Each node represents a segment of the articulated figure, for example, a forearm. For human characters, most of these transforms define rotations (often specified by Euler angles) at joints of the figure. At any moment in time, the pose of a character may be defined by a list of numbers defining its joint angles. A sequence of such poses defines the movement of the articulated figure, and is sometimes referred to as an animation curve. Rag doll motion simulation can be used to produce such an animation curve. So can manual key frame animation or motion capture. It may therefore seem that the combination of different motions may be simulated by combining corresponding animation curves.

Indeed, combination of animation curves is known in the art, as follows: two poses of the same figure from different animation sequences may be combined by interpolating between pose parameters at each frame. For example, if pose "A" has angles 45 degrees and 90 degrees for the left and right elbows and pose "B" has 15 degrees and 70 degrees for the elbows, then a pose half-way between would have an angle of 30 degrees for the left elbow and 80 degrees for the right elbow. Traditionally, animation sequences have been combined in this way by performing an interpolation for each frame (or at selected key frames) of the animation sequences. The prior art includes other, more sophisticated interpolation methods besides simple averaging, for example, weighted averaging. Regardless, interpolation often results in blended

animation sequences that are not as realistic as desired, and sometimes even depicts movements that are obviously impossible.

It is desirable, therefore, to provide a method for defining a combination of differently-defined animation sequences for an articulated figure, that achieves realistic results without requiring an animator or actor to manually account for involuntary motion. It is further desirable to provide a method whereby a given animation sequence depicting a predefined motion can be combined with different simulated involuntary responses, without requiring additional labor-intensive operations, such as manual definitions of key frame poses or motion capture.

SUMMARY OF THE INVENTION

The present invention provides a method for animating articulated figures, that overcome the limitations of the prior art. The method may be used to combine any defined animation sequence of an articulated figure with modeled involuntary movement governed by equations of motion, for the same figure. Applications for the new method may include animating action sequences for motion pictures or computer games, or for any other animation application. The method may be used with standard animation methods and tools used to provide character animation tied to an underlying articulated figure comprising a plurality of segments or nodes linked by defined transforms or joint parameters. Articulated figures of this type, and related animation methods, are well known in the art. Usually, articulated figures comprise data structures having the topography of a directed acylic graph, although the invention is not limited to articulated figures of this type. As is known for articulated figures used in simulated motion, each segment of the articulated figure should additionally have associated parameters defining its mass and center of mass, and a defined angular velocity vector (which may be zero).

Essentially, a method according to the invention may be used to process several differently-defined animation drivers, to thereby obtain an animation sequence that blends both drivers. A first driver may comprise a defined animation sequence for the articulated figure. The defined animation sequence may be manually generated by any

suitable method, or generated in any other desired manner, such as by motion capture. In an embodiment of the invention, the defined animation sequence is intended to depict voluntary movements of an animated character. The animation sequence defines a pose of the articulated figure at defined times of the sequence. The defined times may be selected at increments less than, equal to, or greater than the frame rate for a desired output animation. At each instant of the sequence, the pose defines a position and orientation of each segment of the articulated figure, according to methods known in the art. Accordingly, as the increment of time is known, the sequence defines an angular velocity and a translational velocity for each segment, for each increment of the animation sequence, except for an initial state. Initial velocities for the articulated figure may be separately defined. Position, orientation, and the rate of change of these quantities may be collectively expressed as $Q(t)$, wherein t represents time. Normally these quantities are defined with respect to a frame of reference for the articulated figure.

A second driver may comprise data defining one or more external forces and/or impulses to be applied to the animated character during the sequence, collectively represented herein by $G(t)$. As the notation suggests, $G(t)$ may vary as a function of time. For example, such forces or impulses may include a time-unvarying gravitational force acting on the center of mass of each segment of the figure, and/or an impulse of time-varying magnitude applied at a defined location of the articulated figure. The external forces $G(t)$ may include both translational forces and torques, defined with respect to the articulated figure. An initial position and orientation of a segment of the articulated figures may also be defined with respect to an external reference system, such as to the scene where the action is to occur.

The first animation driver is received as input for an inverse-dynamics solution process. The inverse-dynamics process calculates a solution $F(t)$, representing time-varying internal torques acting on each segment of the articulated figure that will result in $Q(t)$. These internal torques may be thought of as exerted by "muscles" of the articulated figure on its respective body segments. In an embodiment of the invention,

externally-applied forces, such as gravity, are not generally considered. This maintains computational simplicity. In other embodiments, the influence of parameters such as the force of gravity and joint friction may be considered. However, for many animation applications, there may be little benefit, and considerably more computational complexity, associated with including such additional parameters in the solution of $F(t)$. Various inverse-dynamic solution methods are known for articulated figures, and any suitable method may be used.

In an embodiment of the invention, a discrete time increment variable Δt is defined prior to the inverse-dynamics solution step described above. This time variable represents a forward-looking interval over which $F(t)$ is linearly defined, for example, $F(t)$ may be assumed constant over the interval. Thus, in an embodiment of the invention, a value for $F(t)$ may be determined from a difference between a pose $P(t)$ determined from a motion simulation and a desired joint orientation $Q(t+\Delta t)$ at a time Δt in the future. In an alternative embodiment, an approximate value for $F(t)$ over this interval may be directly determined from $Q(t+\Delta t)-Q(t)$, independently of $P(t)$.

In either case, torque may be determined from the relation $\tau = \dot{L}$, where τ represents torque and $\dot{L}$ represents the time derivative of angular momentum. In turn, $\dot{L}$ may be approximated from $I\frac{\omega(t+\Delta t)-\omega(t)}{\Delta t}$, wherein I represents the moment of inertia of the respective segment, and ω is the associated angular velocity. The smaller the value of Δt, the more accurate the approximation for $F(t)$ will be. Generally, $Q(t)$ is defined at discrete time increments (although it may be interpolated between it minimum increments), and in such cases Δt should be defined as equal to, or greater than the minimum time increment of $Q(t)$.

A forward-dynamics solution process may then proceed, starting from an initial state Q_0 of $Q(t)$, at an initial time $t = 0$. Initially, a force $F(t)$ for the time interval from t = 0 to t = Δt may be determined from $Q(t+\Delta t)-Q(t)$. In an embodiment of the invention, the external force function $G(t)$ and $F(t)$ for the first interval may then be

provided as input to any suitable motion simulation process for an articulated figure, calculating an initial pose $P(n\Delta t)$, where n=1. This initial pose may then be compared with $Q(t+(n+1)\Delta t)$, i.e., the desired pose a time Δt in the future, to compute the internal force function $F(t)$ for the interval from Δt to $2\Delta t$. The sum $F(t)+G(t)$ is then provided to the motion simulator for the next interval (n=2) to compute an output pose for this next interval. The foregoing process is repeated until the pose sequence $P(t)$ for the articulated figure is defined over the entire period of interest. The output $P(t)$ may then be used to develop a finished animation according to methods known in the art.

In an alternative embodiment, the internal torque function $F(t)$ may be calculated ahead of time, based on $Q(t+\Delta t)-Q(t)$ independently of $P(t)$. The sum of the internal and external forces $F(t)+G(t)$ may then be provided as input to any suitable motion simulation process for an articulated figure. The motion simulation thereby computes a resulting time-varying pose $P(t)$ for the articulated figure, including its time-varying position and orientation with respect to an external frame of reference, over the entire time period of interest.

As an alternative to driving the forward-dynamics solution process with the sum $F(t)+G(t)$ according to either of the foregoing embodiments, a modified sum defined as $sF(t)+G(t)$, wherein s is a user-selected scale factor, may be used instead. Selecting a value for s less than one will result in a $P(t)$ in which movement corresponding to $F(t)$ is diminished. Setting s greater than one will exaggerate these movements. By adjusting the value of s, a user may adjust the prominence of the separately-defined animation sequence $Q(t)$ in the output sequence $P(t)$, depending on the intended effect. Optionally, s may be defined as a time-varying function.

Advantageously, the output $P(t)$ is a function of the separately determined inputs $F(t)$ and $G(t)$. Any particular animation sequence $Q(t)$ may be combined with any number of separate external force functions $G(t)$, as desired. It should be possible,

therefore, to reduce the need to perform labor-intensive operations for defining $Q(t)$, such as manual key frame animation or motion capture, using a method according to the invention. At the same time, quite varied yet realistic animation results may be realized by combining the same internal force function $F(t)$ with different external force functions $G(t)$. This may readily be accomplished in real time, for application in computer game applications.

Thus, a character animated according to the invention may be provided with a state of motion that depends in part on the character's voluntary movements. No longer will the character behave like a passive "rag doll" under the influence of external forces. For example, if a character pulls its limbs in towards its center of mass while tumbling, its overall angular velocity will increase. In other words, it will tumble faster. Such effects may create an impression than a character is alive and able to influence its motion in a realistic way by voluntary movement, thereby increasing the interest and appeal of action sequences and other animations.

A more complete understanding of the method according to the present invention will be afforded to those skilled in the art, as well as a realization of additional advantages and objects thereof, by a consideration of the following detailed description of the preferred embodiment. Reference will be made to the appended sheets of drawings, which will first be described briefly.

BRIEF DESCRIPTION OF THE DRAWINGS

Figs. 1A are diagrams showing exemplary successive poses of an articulated figure for use in computer animation.

Fig. 2 is a flow chart showing exemplary steps for determining movements of an articulated figure, according to an embodiment of the invention.

Fig. 3 is a block diagram showing an exemplary system for performing a method according to the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention provides a method an system for determining movement of an articulated figure, such as may be used for computer-generated animation, that overcomes the limitations of the prior art. In the detailed description that follows, like element numerals are used to denote like elements appearing in one or more of the figures.

Fig. 1A shows an exemplary articulated figure 100 for use in computer animation. Fig. 1B shows the same figure 100 in a different pose, as it might appear at a later time in an animation sequence. Various forms of articulated figures are known in the art. It should be appreciated that the 3-segment, 2-jointed figure 100 shown in Fig. 1A, although sufficient for illustrating the methods of the invention, is much simpler than a typical figure used for animating humans and the like. Figure 100 comprises "n" number of body segments 101, 102, 103, connected by "n-1" joints 104, 105. A mass and center of mass 106, 107, 108 are associated with each body segment. Joints 104, 105 may be defined with various constraints. For example, joint 104 may be defined as a ball-and-socket joint, while joint 105 may be defined as a pin joint, permitting rotation around a single axis. The connectivity of the joints of an articulated figure having "n" joints and "m" segments may be expressed by a joint connectivity matrix, here denoted $\mathbf{J}$. Articulated figures for animating humanoid or other character movement, and methods for mathematically expressing and calculating poses of such articulated figures, are well known in the art.

For example, three Euler angles, sometimes called joint angles, may be used to define the orientation of an attached segment. In the alternative, nine Euler parameters (direction cosines) may be used to define the orientation. When nine parameters are used, they may be expressed as a 3x3 matrix, herein denoted $\mathbf{R}$. Each segment may be described in its own coordinate space, with relationships between coordinate spaces defined by any intervening Euler angles and segment geometry. A database or table of Euler angles, Euler parameters, or other values which define the position of an articulated figure at successive instants of time is sometimes referred to as an

animation curve. An animation curve that is defined by manual key frame animation, by capturing the motion of an actor or other subject, or in any other suitable manner, is herein referred to as $Q(t)$.

For purely manual, intuitive development of animation curves, it is not necessary to include a mass or center of mass for the articulated segments. Articulated figures with associated mass and center-of-mass parameters are known, however, for simulating the motion of an articulated figure in reaction to externally-applied forces. Software tools are commercially available to perform motion simulation of this type, sometimes referred to as "rag doll" simulation. For example, a rag doll simulator is available, as more fully described at the Internet address www.havok.com. Simulators of this type calculate the reaction of an articulated figure having a defined mass and joint parameters, using Newton's equations of motion. It should be appreciated, therefore, that implementation of the invention does not require any further definition of the animation subject's articulated figure than would be required for a conventional motion simulation.

In the present invention, mass and center-of-mass parameters are used differently than in conventional motion simulation, for the solution of an inverse-dynamics problem: determining internal torques (or if relevant, translation forces) by which an articulated figure can be made to respond as defined by an animation curve. For each segment of the articulated figure, a rotational inertia tensor $\mathbf{I}$ may be defined such that each segment's angular momentum is given by $\mathbf{L} = \mathbf{I}\omega$, wherein ω is the segment's angular velocity tensor. In an inverse-dynamic solution step of the invention, a matrix of inertia tensors $\mathbf{I}$ for each segment may be used to solve for internal torques τ acting at each joint of the articulated figure, so as to result in animation curve that is equivalent or closely corresponding to $Q(t)$. For example, referring to Fig. 1A, a first torque 109 and a second torque 110 may be computed that, when applied over a defined interval of time in the absence of externally applied forces, will move figure 100 from the pose shown in Fig. 1A to that shown in Fig. 1B.

As noted above, for many or most articulated figures, a solution for τ is sufficient to define an animation curve, and definition of translational forces exerted between joints is not required. Most articulated figures modeled after natural beings do not contain joints that permit appreciable translation between jointed segments. Any internal translational forces exerted by the muscles therefore do not perform appreciable work, and may usually be ignored. In the unusual case where the articulated figure includes one or more joints that permits appreciable translation between adjacent segments, non-zero translational forces may also be considered.

The foregoing principles may be applied to determine movement of an articulated figure, including both "voluntary" and simulated involuntary movement, according to the invention. Fig. 2 shows exemplary steps of a method 200 for determining such movement. At step 202, a pose sequence $Q(t)$ for the articulated figure is accessed, such as by reading from a computer database or memory. The pose sequence may be in any suitable form, such as a standard animation curve. Generally this animation curve should depict the figure's voluntary motions (e.g., jumping, running, kicking) and should not attempt to depict involuntary motion from externally-applied forces. Methods for defining such animation curves are well understood in the art. An initial rotational velocity for the segments, $Q_0(t)$, may also be defined. Note that segment velocities subsequent to the initial velocity may be determined from the initial velocities, the positions of the segments at future times as defined by $Q(t)$, and the topology of the articulated figure, such as described by a connectivity matrix $\mathbf{J}$.

In respect to step 202, an amount of time between poses of the animation curve may be defined as a "look-ahead" interval Δt. The look-ahead interval may be user-defined, or may be determined from other parameters of the desired animation, such as the frame rate. A difference $Q(t+\Delta t)-Q(t)$ may be used to define a rotational velocity ω of each segment of the articulated figure, at successive instants of the animation curve. In particular, in each pose of the sequence, an orientation of each joint "i" of the figure may be defined by

$$\mathbf{O}_i = \mathbf{R}_j^T \mathbf{R}_i, \qquad \text{(Eq. 1)}$$

where $\mathbf{O}_i$ represents the orientation of the *i*th joint, and $\mathbf{R}_j$ and $\mathbf{R}_i$ are the orientation of its adjoining joints. An incremental change in $\Delta\mathbf{O}_i$ corresponding to the look-ahead interval, divided by Δt, may represent a segment velocity ω. This, in turn, may be used to solve for the torque τ at each joint, using the basic relationship

$$\dot{\mathbf{L}} = \tau, \qquad \text{(Eq. 2)}$$

where $\dot{\mathbf{L}}$ is the time derivative of angular momentum $\mathbf{L}$, defined as $\mathbf{I}\omega$.

At step 204, such calculation should proceed. Analytically, the determination of the time derivative of $\mathbf{I}\omega$ is not trivial, but the literature contains methods for an efficient solution. The problem may be characterized as the solution of

$$\Delta\mathbf{R} \cong \mathbf{J}^T\mathbf{M}^{-1}\mathbf{J}\tau \qquad \text{(Eq. 3)}$$

for τ, where $\Delta\mathbf{R}$ is a matrix of changes in joint orientation for a given interval, $\mathbf{M}^{-1}$ is a diagonal matrix of inverse inertia tensors for each segment of the articulated figure, and $\mathbf{J}$ and τ are as previously described. Similar problems and solution methods are described, for example, in *Linear-Time Dynamics using Lagrange Multipliers*, by David Baraff, Siggraph 1996 (© 1996 ACM-0-89791-746-4/96/008), and in particular, section 7.3 therein, by which a solution for τ may be obtained in linear time. A collection of values for τ defined over the intervals of interest, may comprise an internal force function $F(t)$, representing the internal torques that will drive the articulated figure to conform to the desired animation curve. As a whole, the algorithm for calculating the internal force function need only be concerned with the joint orientations, and need not be concerned with the overall position or velocity of the segments except as used to calculated the joint orientations.

A target animation curve for defining the internal force function $F(t)$ may be defined in various way. In an embodiment of the invention, the target animation curve is defined in a stepwise fashion, contemporaneously with the calculation of $P(t)$ during motion simulation, as outlined in step 214. During forward-dynamics solution process 214, $F(t)$ for a first time interval from t = 0 to t = Δt may be determined from

$Q(t+\Delta t)-Q(t)$. The external force function $G(t)$ and $F(t)$ for the first interval may then be provided as input to any suitable motion simulation process for an articulated figure, calculating an initial pose $P(n\Delta t)$, where n=1. This initial pose may then be compared with $Q(t+(n+1)\Delta t)$, i.e., the desired pose a time Δt in the future, to compute the internal force function $F(t)$ for the next interval. For example, for n=2, the next interval is Δt to $2\Delta t$. The sum $F(t)+G(t)$ for the next interval may then be calculated as described for step 214 to compute a corresponding output pose. The foregoing may be repeated until the pose sequence $P(t)$ for the articulated figure is defined over the entire period of interest. Optionally, if a particular value of $F(t)$ exceeds a specified value for a muscular torque at any particular joint, it may be limited to the specified maximum value. Such constraints may be derived from biometric data, and may be used to prevent unrealistic character movement, if desired.

In an alternative embodiment, the internal force function may be calculated directly from $Q(t)$, independently of the calculation of $P(t)$ in step 214. In such case, $F(t)$ may be calculated from $Q(t+\Delta t)-Q(t)$ for each successive interval. The sum of the internal and external forces $F(t)+G(t)$ may then be provided as input to any suitable motion simulation process for an articulated figure. Note that this alternative method should generally result in a different output $P(t)$ for a given $Q(t)$ and $G(t)$, as compared to the first embodiment described above. Because there is no feedback between $F(t)$ and $P(t)$ according to this second embodiment, the predetermined component of the animated movement may "drift" from its original appearance as dictated by $Q(t)$. Whether or not drift is desirable may depend on the intended effect.

The invention is not limited to a particular solution method. In more general terms, step 204 may be described as determining those internal forces of the defined articulated figure that will drive the figure according to its predefined animation curve. A variety of different approaches may be suitable for solving this problem. Generally, external forces, including the force of gravity, may be ignored at this step, for computational simplicity. It should be appreciated that the object of step 204, unlike

many prior-art applications for inverse dynamics, is not generally concerned with an accurate solution for a robotics or biomechanical problem. Instead, a principal object of step 204 is to transform an arbitrarily-determined animation curve to a form that may readily be combined with other input to a motion-simulation function. Hence, the force function $F(t)$, whether defining tensors τ and/or other data, should be configured for this purpose.

At step 206, data $G(t)$ may be accessed, such as from a computer database or memory. $G(t)$ may be defined in an early step, and should define one or more external forces and/or impulses to be applied to the articulated figure during a desired animation sequence. Data $G(t)$ may include values defining both translational forces and torques, defined with respect to the articulated figure. Such forces may include, for example, a force of gravity, forces from interactions with flowing fluids, and impulse forces from collisions with other objects. Such forces may be scripted, and/or defined using a computational method, such as collision detection. Motion of the articulated figure is to be simulated starting from some defined initial position and state. Therefore, an initial position and orientation of a segment of the articulated figures may also be defined with respect to an external reference system, such as to the scene where the action is to occur. The initial pose of the figure may be defined by $Q(t)$.

At optional step 208, a scale factor s may be determined. The scale factor may be used to scale the component of simulated motion that is caused by the internal force function $F(t)$. The scale factor may be constant, or a time varying function $s(t)$. For example, a constant value less than one may be used to reduce the amount of voluntary character movement during an action sequence; a constant value greater than one may be used to exaggerate this component of motion; and a time-decaying value may be used to gradually reduce the component of voluntary character movement. A user interface, such as a slider, may be used to receive direct user input. At step 210, if a scale factor is entered, the solution $F(t)$ to step 204 may be appropriately scaled by

the selected amount. For example, each tensor τ comprising $F(t)$ may be scaled by s. If no scale factor is specified, $F(t)$ may be left unscaled.

At step 212, the force functions previously defined may be summed. If no scale factor is to be used, the resulting sum may be represented by $F(t)+G(t)$. If a scale factor is to be applied, the resulting sum may be represented by $sF(t)+G(t)$. In either case, the sum should be expressed in a form suitable for providing input to a motion simulation program for articulated figures. It should be appreciated that in some cases, the force functions may be separately provided to a motion simulation algorithm, and implicitly summed though iterative motion simulations. For example, an intermediate simulated result may be calculated using $G(t)$ as input, and then the intermediate result and $F(t)$ may be provided as input to a second iteration of the motion simulation to obtain a final result. The final result may be the same as if the sum $F(t)+G(t)$ had been provided to the motion simulation algorithm. Such techniques merely perform the same summation implicitly, and should be considered as within the scope of the invention.

At step 214, the sum computed at step 212 is used as input to a motion simulation algorithm, which determines a simulated motion $P(t)$ for the articulated figure using a forward-dynamics solution process. Essentially, the equations of motion are used to define the position of each segment of the articulated figure at each desired time increment. Such calculation methods are straightforward and need not be described here. For example, a rag doll motion simulation algorithm may be configured to operate on the sum of $F(t)+G(t)$. Any suitable method motion simulation method as known in the art for articulated figures may be used. The resulting output $P(t)$ may provide the basis for a character animation, as known in the art. As previously described, the output $P(t)$ for each time increment may also be compared with the predetermined animation curve $Q(t)$ overall a look-ahead interval Δt to calculate $F(t)$ during step 214, thereby driving the output animation towards a desired pose.

The methods of the invention may be performed using any suitable computer system. Fig. 3 shows a block diagram of a computer system 300, comprising a computer 302 connected to a memory 306 containing instructions for performing steps of the invention, and to a database 304 containing previously-defined data, such as an input animation curve $Q(t)$ and/or an external force function $G(t)$. Computer 302 may be configured to read a removable media 308, on which instructions for performing the steps of the invention may be encoded. Such instructions may be read and transferred to memory 306 for execution by computer 302. The instructions may be written in any suitable programming language and encoded for performing steps of a method according to the invention, as described above.

Having thus described preferred embodiments of a method for animating an articulated figure, it should be apparent to those skilled in the art that certain advantages of the within system have been achieved. It should also be appreciated that various modifications, adaptations, and alternative embodiments thereof may be made within the scope and spirit of the present invention. The invention is defined by the appended claims.